

WE CLAIM:

1. A computer-readable medium having computer-executable components, comprising:
a server component configured to receive from a client information that indicates the client needs additional resources to perform a transaction, the server component being further configured to determine if allocating to the client the additional resources puts the server component in a resource constrained situation, and if so, to rebalance resources currently allocated to a plurality of existing clients.
2. The computer-readable medium of claim 1, wherein the server component executes on a server in a network environment.
3. The computer-readable medium of claim 1, wherein the server component is further configured to allocate the client the additional resources needed if the server determines that such allocation does not create the resource constrained situation.
4. The computer-readable medium of claim 1, wherein the clients and the server component communicate using a light weight input/output protocol.
5. The computer-readable medium of claim 1, wherein the server component determines if the resource constrained situation occurs by comparing a current number of resources allocated to the client with a total number of available resources.
6. The computer-readable medium of claim 5, wherein the determination further comprises comparing a current number of resources allocated to every client connected to the server component with the total number of available resources.
7. The computer-readable medium of claim 6, wherein the determination further comprises comparing the current number of resources allocated to every client

connected to the server component and a number of requested resources with the total number of available resources.

8. The computer-readable medium of claim 1, wherein the rebalance of the resources is performed based on an equitable distribution of the resources among the plurality of clients.

9. The computer-readable medium of claim 8, wherein the equitable distribution of the resources is based on a number of clients connected to the server component.

10. The computer-readable medium of claim 9, wherein at least one client connection has a preferential weighting with respect to other clients.

11. The computer-readable medium of claim 8, wherein the equitable distribution of the resources is based on a number of open files associated with each client connected to the server component.

12. The computer-readable medium of claim 11, wherein at least one open file has a preferential weighting with respect to other open files.

13. A computer-readable medium encoded with a data structure, comprising:
a plurality of data stores, each data store being associated with a different client connection to a server, each data store including:
a credits used field that identifies a number of resource credits currently in use by a client corresponding to the data store;
a credit limit field that identifies a number of resources available to the client corresponding to the data store;
a pending count field that identifies a number of transactions that are pending due to an unavailability of sufficient resources to handle the transactions; and
an open files field that identifies a number of files that are currently in use by the client.
14. The computer-readable medium of claim 13, wherein the data store further comprises a flag field that identifies whether the corresponding client has acknowledged a resource-related message.
15. The computer-readable medium of claim 13, wherein a value of the pending count field is provided by the client in connection with a transaction request message.
16. The computer-readable medium of claim 15, wherein a value of the credit limit field is modified based on the value of the pending count field.
17. The computer-readable medium of claim 13, wherein values for the credit limit fields of the plurality of data stores is rebalanced based on an equitable distribution of available resources.

18. A computer-readable medium encoded with a data structure, comprising:
- a credits used field that identifies a number of resource credits currently in use by a client corresponding to the data structure;
 - a credit limit field that identifies a number of resources available to the client;
 - a pending count field that identifies a number of transactions that are pending due to an unavailability of sufficient resources to handle the transactions; and
 - a pending queue field that includes transaction messages corresponding to the transactions that are pending.

19. A computer-implemented method, comprising:
 computing a total number of client connections, each client connection being associated with a client connected to a server, each client having a credit limit that identifies a number of resources that are allocated to the client
 computing a total number of pending requests that identifies a number of transaction requests that are not being handled due to a limitation on resources;
 computing a total number of credits in use; and
 if the total number of pending requests and the total number of credits in use combined exceeds a total number of available resources, calculating a new credit limit for each of the clients connected to the server; and
 reallocating the total available resources in accordance with the new credit limits.

20. The computer-implemented method of claim 19, wherein the reallocation is based on each client connection receiving a pro rata share of the total available resources.

21. The computer-implemented method of claim 20, wherein the pro rata share of the total available resources is based on the total available resources divided among the total number of client connections.

22. The computer-implemented method of claim 21, wherein the total available resources are divided evenly among the total number of client connections.

23. The computer-implemented method of claim 21, wherein at least one of the client connections is weighted more heavily than another of the client connections.

24. The computer-implemented method of claim 20, wherein the pro rata share for a particular client is based on a proportion of a total number of open files to a number of open files for the particular client.